Lab4Schools

Lab Activity “ERMET”

Co-funded by the Erasmus+ Programme of the European Union

ERMET

TIME: 4 h

GOAL

You have to be able to :

- Create a GRAFCET order from a set of specifications then download it to an automaton .

WHAT YOU HAVE TO WORK ON

- Create an adressing input and output table.
- Create a program with « SoMachine »
- Transfer this program.
- Test the program.

TOOLS NEEDED

- Computer with the software « SoMachine »
- The model of the ERMET
- « SoMachine » instructions booklet.

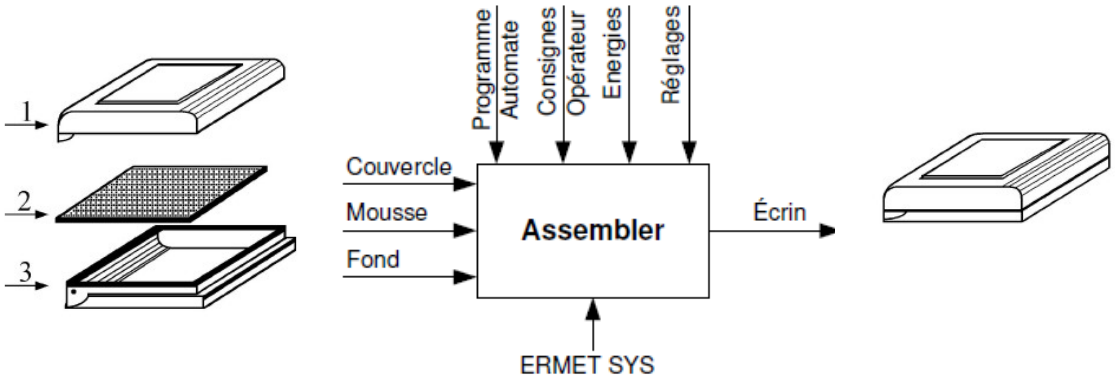
PRESENTATION

- The ERMET system is a production system extracted from a jewelry box packaging machine.

As configured, the machine is used to assemble jewel cases to receive costume jewelry on a packaging line.

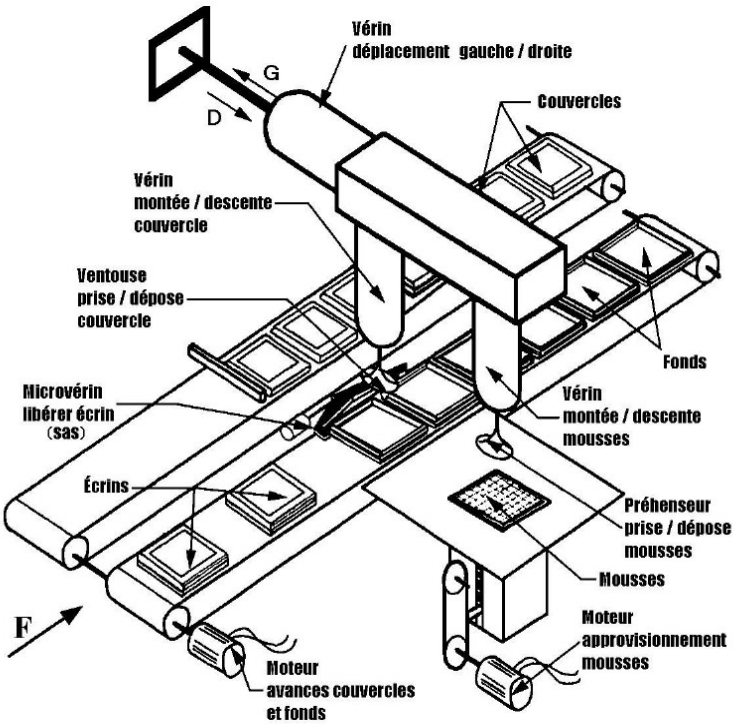
- **Main work material and modeling.**

The system receives three parts (1 the lid, 2 the foam and 3 the bottom) before the assembly operation. These three parts form the case.



- **Architecture of the packaging system.**

The conditioning system has :
- A belt for the arrangement of the bottoms. This same belt is also used for the evacuation of finished cases.
- A conveyor belt for the lids.
- A device for gripping the lids. This same device is used to to fit the lids on the bottoms.
- A device for gripping the foams. These foams are stored in a a magazine.
- A transfer device allowing :
  - The transfer of the lids on the bottoms.
  - The transfer of the foams on the bottoms.



- **Operation of the conditionning system:**

Station n°1 : Lid gripping (Cylinder 4C)

A pneumatic cylinder equipped with a suction cup comes to seize a lid on the belt of arrangement and emboîte it on a bottom at the level of the belt of arrangement of the funds.

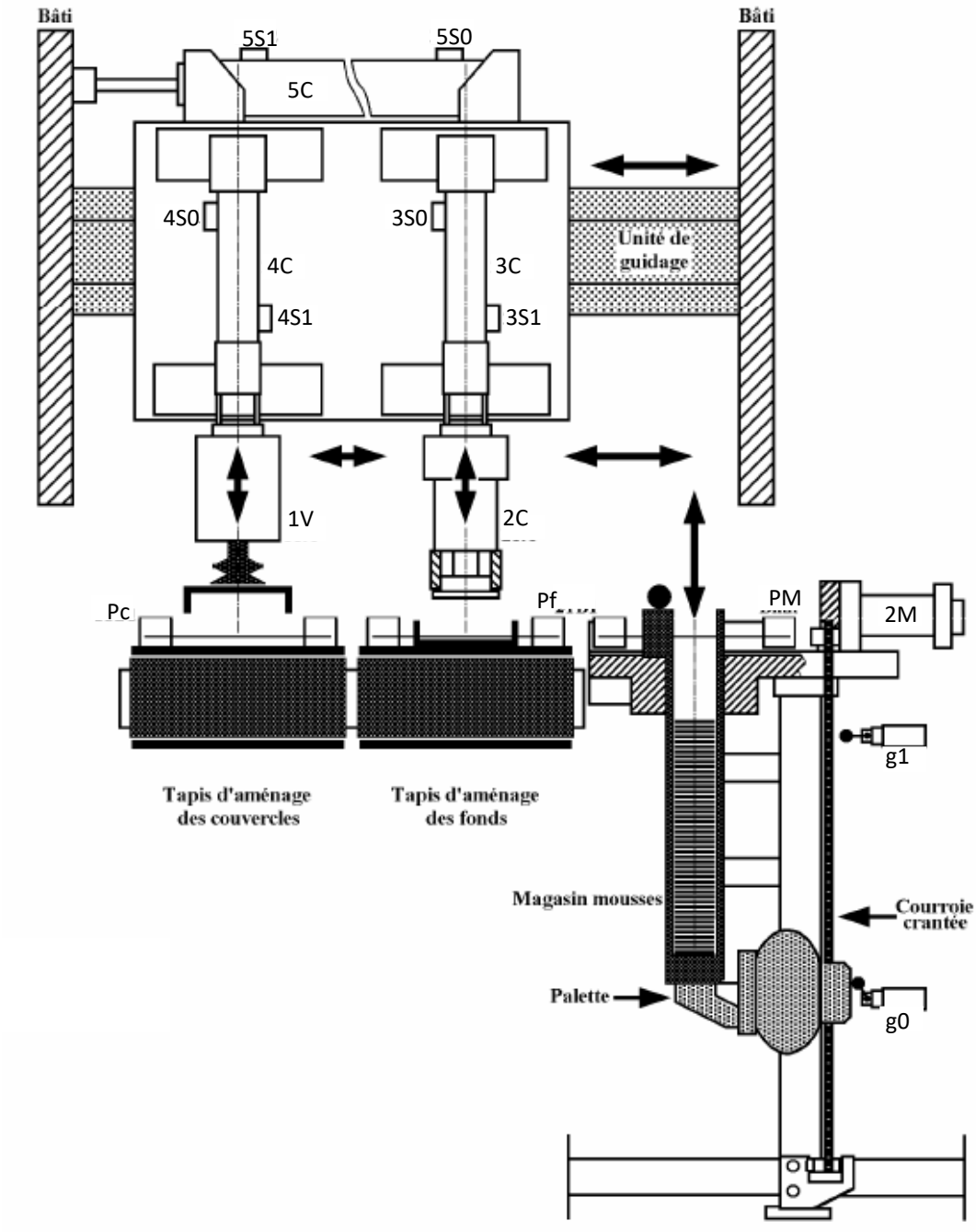
Station n°2 : Foam gripping (Cylinder 3C)

A cylinder equipped with a gripping device picks up a foam filling stored in the "foam magazine" and places it in a bottom at the level of the bottom fitting belt.

Station n°3 : Transfer device. (Cylinder 5C)

A linear translation unit driven by a pneumatic cylinder allows the transfer of the foams in the bottoms and the transfer of the lids on the bottoms.

.



- **Nomenclature :**

Taches	Actionneurs		Ordre Pré-actionneurs			Informations Capteurs/IHM			
	Type	Code	Type	Code	Adresse	Informations	Type	Code	Adresse
bouton poussoir								Départ cycle	%IX0.0
bouton coup de poing								Arrêt d'urgence	%IX0.6
bouton poussoir								initialisation	%IX0.3
bouton poussoir								Réarmement	%IX0.5
Sélecteur 3 positions								coup/coup	%IX0.2
								auto	%IX0.1
Sélecteur 3 positions avec rappel en position neutre								Chariot Gauche	%IX0.7
								Chariot / Droite	%IX1.1
bouton poussoir								Validation magasin mousse	%IX0.4
Voyant cond,initiales				H7	%QX0.6				
Voyant arrêt d'urgence				H6	X				
Voyant manque mousse				H5	%QX0.7				
voyant sous-tension				H4	X				
voyant auto				H3	%QX0.4				
voyant coup/coup				H2	%QX0.5				
voyant arrêt				H1	%QX1.3				

Nomenclature des taches **Projet: ERMET**

Taches	Actionneurs		Ordre Pré-actionneurs			Informations Capteurs/IHM			
	Type	Code	Type	Code	Adresse	Informations	Type	Code	Adresse
Descendre/monter préhenseur mousse	vérin double effet	3C	distributeur 5/2 monostable à commande électrique	3D	%QX3.1	Tige Sortie (bas)	capteur ILS	3S1	%IX2.0
						Tige rentrée (Haut)		3S0	%IX2.1
Préhension mousse	vérin double effet (pince)	2C	distributeur 5/2 bistable à commande électrique	2D+	%QX1.5				
				2D-	%QX3.0				
Monter/descendre préhenseur couvercle	vérin double effet	4C	distributeur 5/2 monostable à commande électrique	4D	%QX3.2	tige sortie (bas)	capteur ILS	4S1	%IX2.3
						Tige rentrée (haut)		4S0	%IX2.2
Préhension couvercle	générateur à vide (Venturi)	1V	distributeur 5/2 monostable à commande électrique	1D+	%QX3.7				
				1D-	%QX1.4				
Déplacement horizontal du bloc	vérin double effet	5C	distributeur 5/2 bistable à commande électrique	5D+	%QX3.5	tige sortie (droite)	capteur ILS	5S1	%IX2.4
				5D-	%QX3.6	tige rentrée (gauche)		5S0	%IX2.5
Vérin Sas	vérin double effet	6C	distributeur 5/2 bistable à commande électrique	6D+	%QX3.3	tige sortie (sas fermé)	capteur ILS	6S1	%IX1.0
				6D-	%QX3.4				
Convoyer fond/couvercle	Moteur Synchrone	1M	contacteur	KMT	%QX1.0	présence couvercle	Capteur à fibre optique à proximité	Pc	%IX1.2
						présence fond	Capteur à fibre optique à proximité	Pf	%IX1.3
Distribution mousse	Moteur courant continue	2M	contacteur	KMMO	%QX1.2	Magasin mousse en haut	Capteur à galet	g1	%IX1.5
				KMDE	%QX1.1	Magasin mousse en Bas	Capteur à galet	g0	%IX1.6
						Présence mousse	Capteur à fibre optique à barrage	PM	%IX1.4
Contacteur de sécurité			contacteur	KMPO	%QX1.6	Contacteur enclenché		KMPO_enclenchee	%IX0.6

Phase 1 : Having a look around at your job

As you can see, the system is already wired.

The control part of your station is composed of a SCHNEIDER M241 programmable logic controller (PLC).

The operating part will only be used in part (1 or 2 actuators and their sensors).

Locate on the station all the elements participating in the action and acquisition chains relating to the actuators used.

Phase 2 : Writing the grafcets

Being interested only in one cylinder, you will program a pendular cycle. That is to say that the stem of the jack will have to leave until the end of race then to return to its initial position. The triggering of the cycle will be done by the activation of the button start cycle " dcy ".

2.1 - After having analyzed the nomenclature of the system, on sheet of copy, to draw up the table of the inputs/outputs (name, mnemonic, assignment API) only of the necessary elements. Note: You will add the output variable "KMPO" of the safety contactor necessary for the operation of the machine.

2.2 - On a copy sheet, write the grafcet point of view PC.

2.3 - With the help of appendix 1, establish the program from your PC Grafcet and your input/output table on the SoMachine software.

2.4 - Call the teacher to upload your program in the PLC.

2.5 - Test your program in the presence of the teacher.

Phase 3 : Including a TEMPORIZATION part

The pendular cycle is the same as before but the rod of the jack will have to go out until the end of race and to remain in position during 3 seconds then to return to its initial position.

3.1 - On a sheet of paper, write the new PC Grafcet.

3.2 - With the help of appendix 2, modify the program from your PC Grafcet on the SoMachine software.

3.3 - Test your program in the presence of the teacher

Phase 4 : Including a COUNTER part

The pendular cycle is the same as before but it must be executed 3 times then stopped until the presence of the cycle start appears again.

4.1 - On the copy sheet, write the new PC Grafcet.

4.2 - Using appendix 3, modify the program from your PC Grafcet on the SoMachine software.

4.3 - Test your program in the presence of the teacher.

Phase 5 : Programming a more difficult cycle (given by the teacher)

5.1 - On a copy sheet, write the new PC Grafcet.

5.2 - Modify the program from your PC Grafcet on the SoMachine software.

5.3 - Test your program in the presence of the teacher.

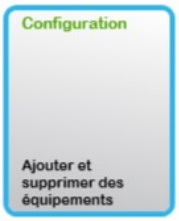
ANNEXE 1 – API Configuration and programming of SoMachine

- **S**tart the computer and launch "SoMachine".

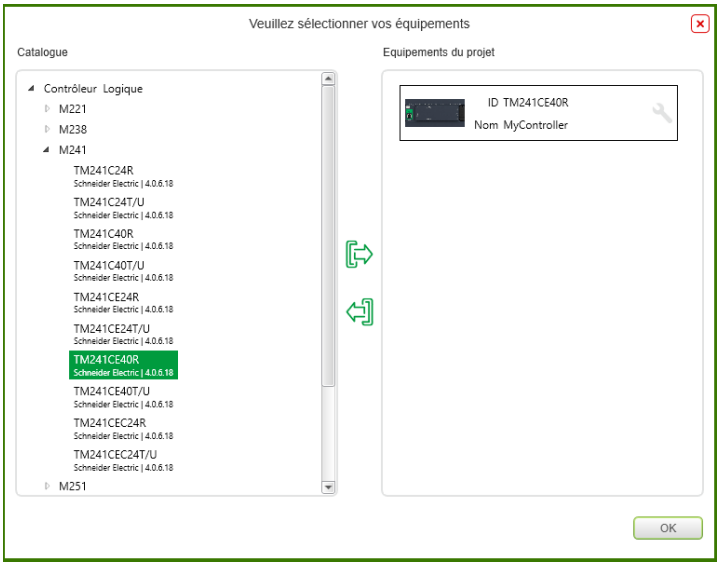
- Create a new project by clicking on the icon "Cretae a project".

- Choose an "empty project".
- Give a name to your project.
  - "OK".

- Double-click on the icon "Configuration".

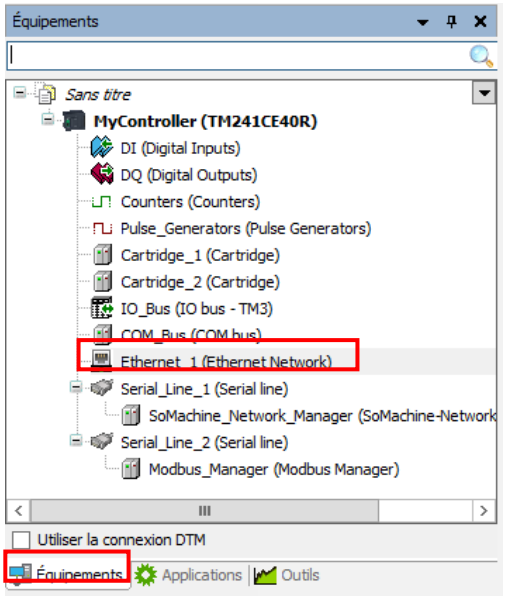


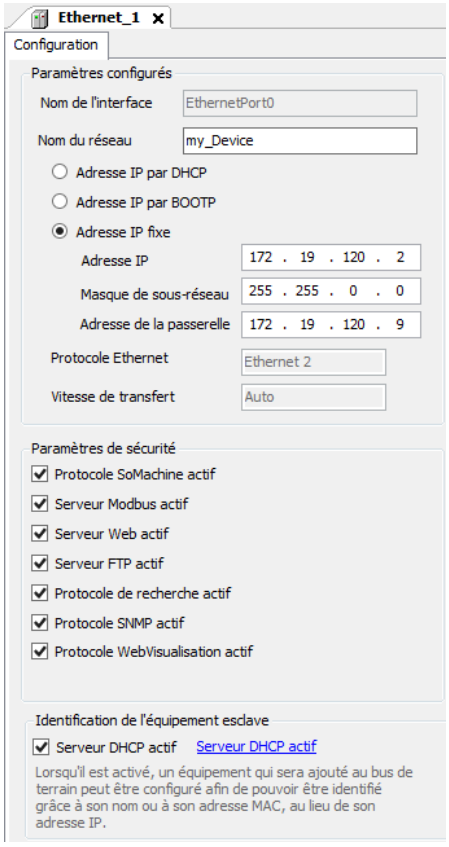
- In the browser "Catalogue", choose your "Contrôleur Logique". The type of processor is given in front of the PLC M241.
- Move the "Contrôleur Logique" in "Equipements du projet" using the arrow.
  - "OK".



- Double-click on "Contrôleur".

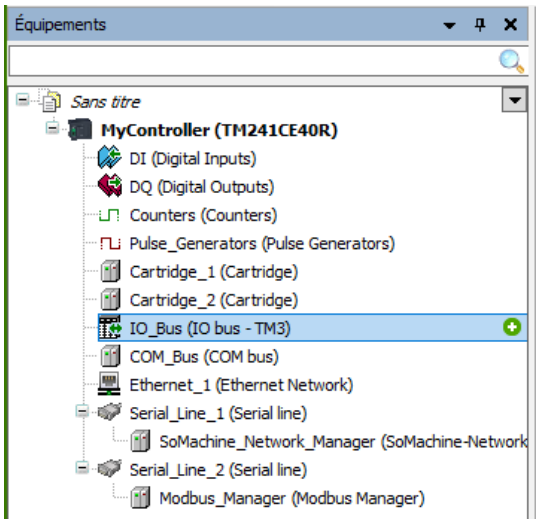
- In the "Equipment" tab, double-click on "Ethernet_1 (Ethernet Network)" :
- Enter the fixed IP address as below in order to communicate with the PLC via an Ethernet port :

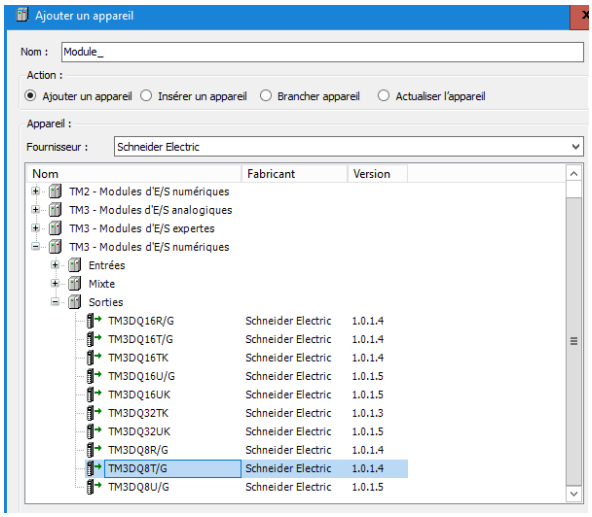


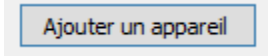


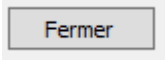
- In the "Equipement" tab, click on the of "IO_Bus (IO bus - TM3)" :

- Choose your digital output extension module. The type of TM3 module is indicated on the front of the module at the PLC.

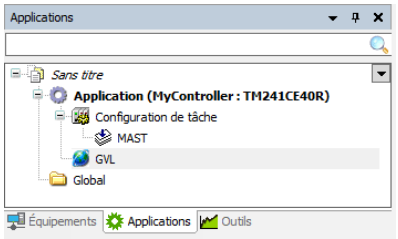




- "Ajouter un appareil / Add a device". 

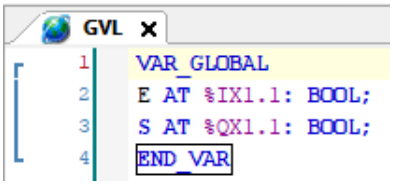
- "Fermer/ Close". 

- In the "Application" tab, double-click on "GVL" :



- Using your I/O table, fill in the assignment of all the PLC addresses of each I/O.
  - For the input variable → "NomVariable AT %IX*.*: BOOL;"
  - For the output variable → " NomVariable AT %QX*.*: BOOL; "

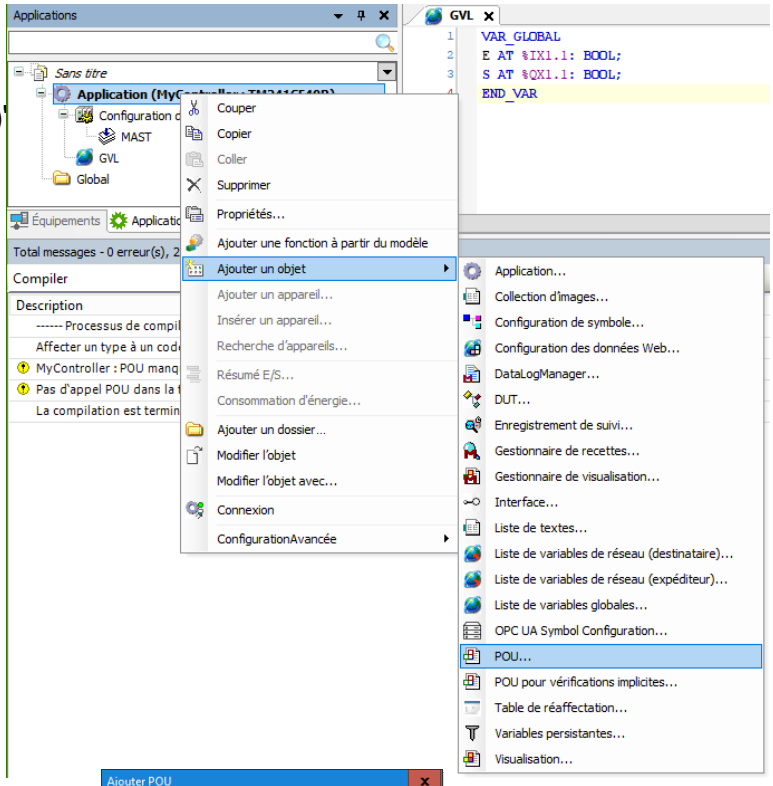
Ex : - The input variable "E" affectée à l'adresse %IX1.1 de l'API.
- The output variable "S" affectée à l'adresse %QX1.1 de l'API.



Make sur the teacher checks your work.

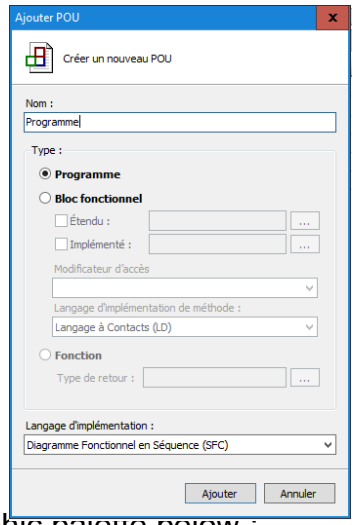
To open a programmation tab:

- Right-click on "Application (MyController :)"
  - "Add an object"
    - "POU"



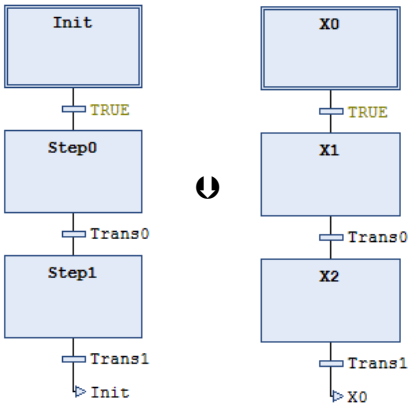
In tbhe window "Add a POU" :

- Write a name.
- Choose "SFC" as implementation language.
  - "Add"



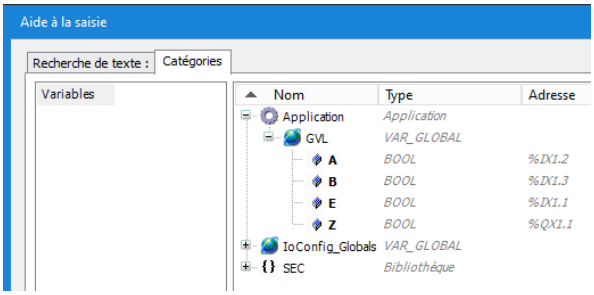
- Enter the skeleton of your Grafcet using the graphic palette below :

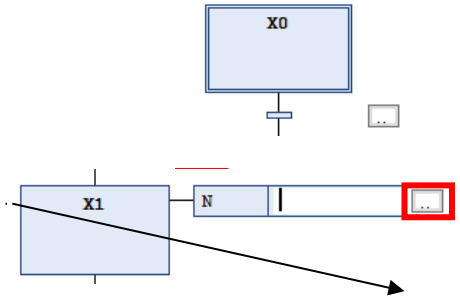
- Rename the steps of your Grafcet by X0, X1, X2... like in the example below :



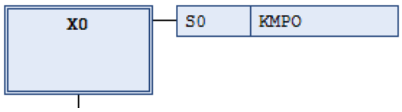
- Enter the receptivities and the actions of your Grafcet.

You can also click on the white rectangle
next to the receptivity or the action and choose the
the desired variable in the "GVL" as below.

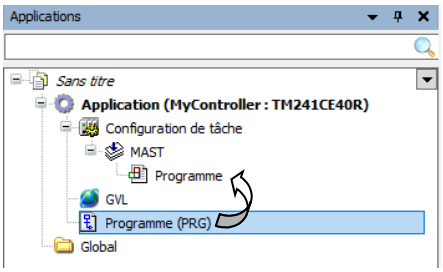




- Enter the setting of variable "KMPO" at step "X0" in order to switch on the safety contactor.



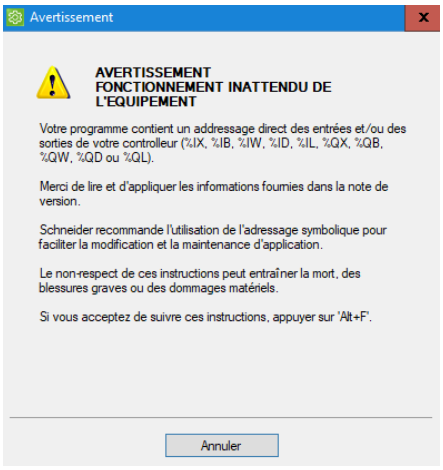
- Drag your program into the "MAST" folder with a long left click so that it can be executed.



- Compile your program by clicking on or on "F11".

- You'll see a warning message that is not important, click on "Alt+F".

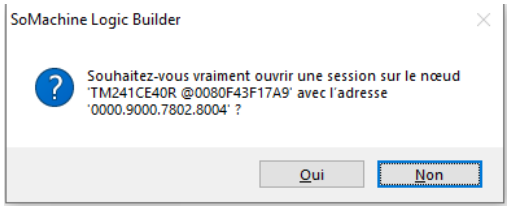
Note : If there's any mistake, correct them and compile your program again.

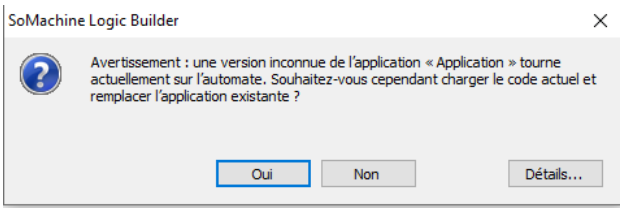


- Connect to the PLC by clicking on or by clicking on "Connexion".

Several messages will appear

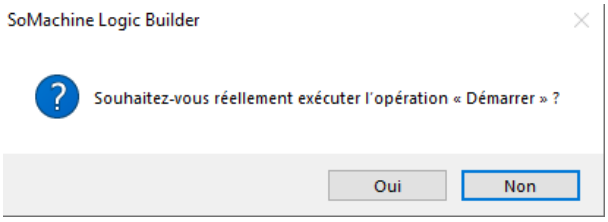
- Click on "Yes"





- Put the PLC on "RUN" by clicking on "Start".

- Click on "Yes"



You can test your program

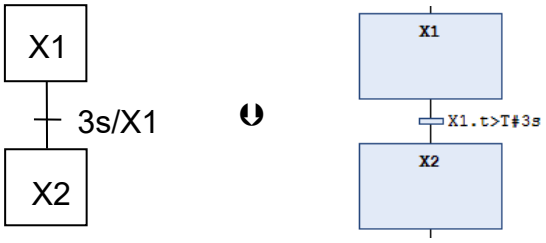
- To stop or make a change to the program, click on "Stop" and then on "Disconnect". .

- Modify your program and redo the steps from compilation to RUN.

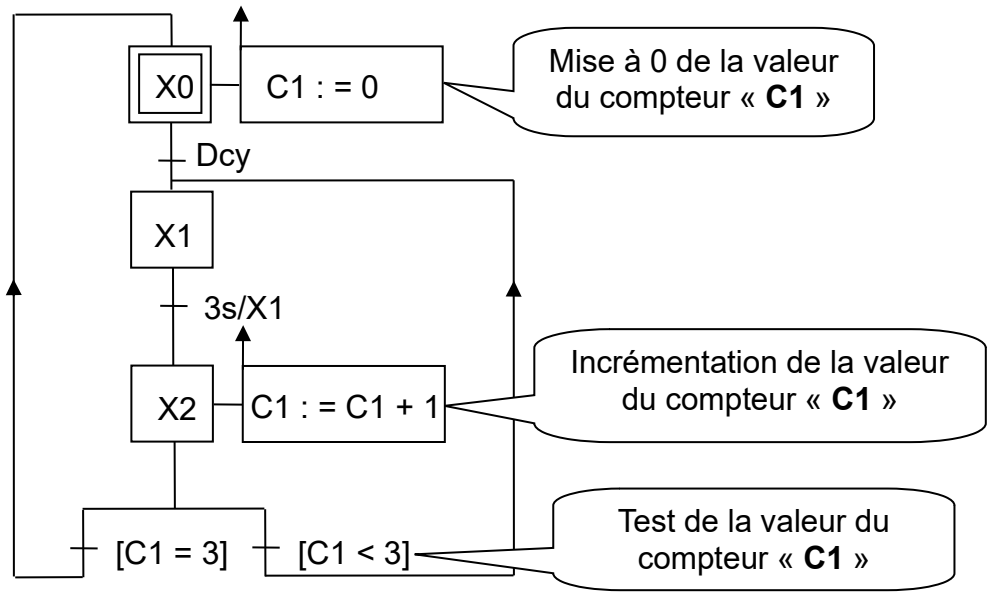
ANNEXE 2 – Ecriture d'une temporisation

Here is an example of a temporization which is triggered at the step X1 of a grafcet and which lasts 3 seconds.
This writing is placed at the level of the receptivity where you test the temporization.

With this method one tests the value of the time which elapses from the moment when the stage X1 was activated "X1.t".
When this value becomes greater than 3 seconds "T#3s", then the receptivity of the transition between step X1 and step X2 becomes true, the transition is validated and therefore can be crossed.

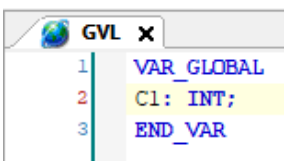


ANNEXE 3 – Ecriture d'un compteur

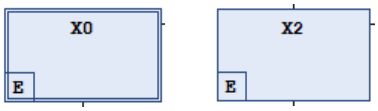


First of all, you have to declare a variable of type "integer" (INT) corresponding to the value of the counter. It can be named as you wish, in the example, this variable is named "C1" for Counter n°1.

- In the "GVL" window, enter the variable "C1" of type "INT" as below:

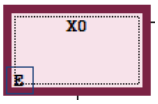


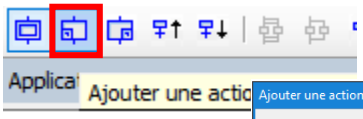
Zeroing "C1:=0" and incrementing the counter "C1:=C1+1" must be done when the step is activated (input action)



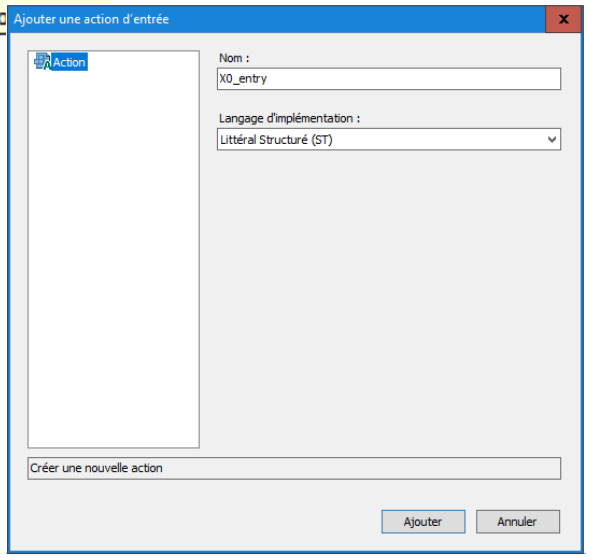
- To set the counter to 0:

🡻 Click on the "X0" step and using the graphics palette, click on "Add Input Action":





The writing of the set to 0 "C1:=0" must be done in the programming language
"Structured Literal" (ST) programming language



In the "Add input action" window:

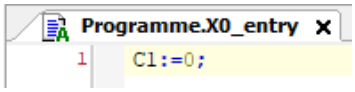
✹ Leave the default name.

✹ Choose as implementation language "ST".

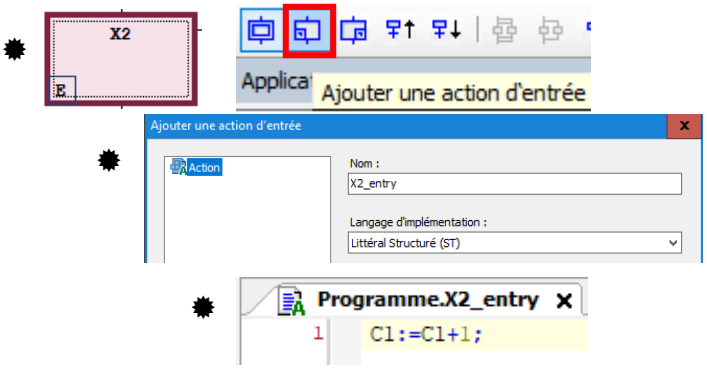
✹ "Add"

In the window :

🡻 Write the equation "C1:=0" in Structured Literal (ST)
Note: an action always ends with a ";".



- For the incrementation of the counter, it is the same method as the setting of the counter to 0:



-Both tests are written in the same way but without the brackets as below:

[C1<3] 🡻 C1<3

[C1=3] 🡻 C1=3